VISUAL ATTRIBUTE LABELING OF IMAGES

A Dissertation
Presented to
The Academic Faculty

By

Varun Agrawal

In Partial Fulfillment
of the Requirements for the Degree
Master of Science in the
School of Interactive Computing

Georgia Institute of Technology

August 2019

VISUAL ATTRIBUTE LABELING OF IMAGES

Approved by:

Dr. James Hays, Advisor
School of Interactive Computing
Georgia Institute of Technology

Dr. Irfan Essa
School of Interactive Computing
Georgia Institute of Technology

Dr. Devi Parikh
School of Interactive Computing
Georgia Institute of Technology

Date Approved: July 30, 2019

Fairy tales are more than true: not because they tell us that dragons exist, but because they tell us that dragons can be beaten.

G.K. Chesterson

Dedicated to Margaret Lobo. A better grandmother one could not ask for.

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my parents, without whose support and encouragement I wouldn't be at this stage. They always believed in me and encouraged me to go to graduate school, and this document is testimony to their belief.

I would like to thank my advisor, Dr. James Hays, for not only taking me on as a student, but for also grounding my research in thorough analysis and helping me think about big picture problems while providing me with tremendous freedom to pursue various ideas.

Finally, I want to thank everyone who has been a major part of my Masters journey. It's been an incredible ride.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

SUMMARY

In this work, we analyze and apply various recent techniques in visual attribute recognition and labeling on a common benchmark dataset in order to motivate the design of a novel framework for this task. Using the large scale COCO Attributes dataset as our benchmark, we systematically investigate recent techniques and advances in the attribute recognition literature in a unified fashion, drawing comparisons and insights from the results. We leverage these insights to propose a new models and loss function to better model the function space of attribute prediction models. Our proposed techniques are based on standard efficient building blocks readily available to researchers and practitioners, are conceptually simple, and are theoretically grounded, while giving state-of-the-art results, and generalises to various sub-domains of attribute recognition. Experiments and ablation studies performed on our model and other methods further corroborate the design decisions for our framework and shed light on possible future avenues of investigation. Our hope is that our model serves as a tool to embed strong visual attribute recognition in more complex visual reasoning tasks and pipelines.

CHAPTER 1
INTRODUCTION AND BACKGROUND

1.1 Introduction

In recent years, computer vision has made phenomenal success in various tasks such as Object and Scene Classification, Object Detection and Human Pose Estimation. Advances in machine learning models have allowed the learning of strong representations and features, allowing researchers to model more complex tasks beyond single labels. One such task is Visual Attribute Recognition, which involves predicting visually relevant attributes to scenes/objects (*e.g. standing* woman). The premise of attribute recognition is that there exists more than a single label that is applicable to the object or scene in consideration, thus making the task strictly more complex than single label classification. Attribute recognition is an important task since good performance in this task allows one to use its results in more complex problem domains such as fine-grained classification, object detection, image search, visual question answering, scene understanding, etc. In this direction, there has been a fair amount of work in the past few years on modeling attribute recognition. However, most of this work has focused on specific domains (such as faces or fashion), making it hard to distinguish and compare the pros and cons of each technique, and thus providing little insight on the benefits achieved by these methods for general attribute recognition for the tasks specified previously.

In this work, we aim to improve our understanding of modern techniques for this task via empirical analysis and experimentation on a single large-scale visual attributes benchmark, thus allowing for their general purpose use, independent of any specific domain. The common benchmark we use is the COCO Attributes dataset [1], a large scale dataset of images with 204 labeled visual attributes. The COCO Attributes dataset has many desirable

properties: it is large scale, hence apt for training powerful deep learning models; it does not assume any specific domain, thus having attribute labels for various types of objects present; and is amenable to precise evaluation via the mean average precision metric. By analyzing various attribute recognition techniques on this common benchmark, we are able to better understand the strengths and weaknesses of each technique in a fair way, allowing us to draw inferences that we then use to propose new loss functions and a new model that alleviates many of the assumptions and weaknesses of the techniques and achieves state-of-the-art performance on this dataset.

Our proposed model, while drawing inspiration from existing models, is primarily based upon the Neural Module Network framework proposed by Andreas *et al*. [2] and further work on Neural Module architectures by Johnson *et al*. [3]. Neural Module Networks are based on the observation that compositionality exists within language structure (*e.g*.a tomato can be red or green) and this compositionality can be learned and leveraged to improve performance on tasks such as Visual Question Answering. We make a similar observation, that visual attributes are compositional by their very nature, and we can use simple, uniform neural modules to learn attribute prediction for each module, simplifying the task of attribute recognition to a binary classification task for each attribute.

Furthermore, as we will see in section 2.1, learning label co-dependencies forms an integral part of visual attribute recognition, especially for performing well on labels which are sparse in the dataset. This is intuitive, since certain attributes are commonly seen together (*e.g*."enjoying" & "smiling", "sleeping" & "laying"), while some are mutually exclusive (*e.g*."angry" & "laughing"). We explicitly learn these co-dependencies by leveraging an additional classifier to embed the outputs of each attribute module into a shared embedding-space, which we then train in a multi-label fashion with different proposed losses. This technique does not assume any prior relationships between modules and attributes, and is easily extensible, without eliminating the possibility of adding in domain knowledge. During inference, we simply query the final classifier and pass the logits through a sigmoid to

obtain binary prediction probabilities of each attribute.

To summarize, we make the following contributions:

1. We establish common baselines by performing a large scale analysis of modern attribute recognition techniques proposed in the literature on a common, challenging benchmark, providing a streamlined approach for future work in this area.

2. We perform an analysis of the strengths and weaknesses of these various techniques in order to leverage their unique capabilities.

3. We propose new loss functions which optimize for approximations of common ranking metrics and thus better model the desired behavior for the attribute labelling task.

4. We propose a novel neural module based attribute recognition architecture which takes advantage of attribute compositionality and co-dependency, as well as insights from our prior analysis, to establish new state-of-the-art in terms of performance on the COCO Attributes dataset. This framework has the added benefits of being easily extensible and fast to train, allowing its use in multiple domains beyond object attribute recognition.

1.2 Related Work

1.2.1 Visual Attributes

Visual attribute recognition has been a long standing problem in Computer Vision and has seen significant research effort in tackling. As a result, a myriad of large scale datasets and techniques for attribute recognition have been proposed over the years, most of which tackle specific sub-domains of images, such as pedestrians, faces, and animals. Russakovsky and Fei-Fei [4] considered learning visual attributes from ImageNet data by utilizing the synset connections between object categories. For pedestrian attribute recognition, much research has been motivated with the intent of improving autonomous vehicles and

surveillance systems. Work from Fukui *et al*. [5] specifically focuses on autonomous vehicles, allowing them to better identify pedestrian behavior. Wang *et al*. [6] specifically focus on human attribute recognition for use in surveillance systems. Their method makes use of an LSTM on multiple parts of an image in order to better correlate attributes with different parts of the human. In the same domain, Li *et al*. [7] use a pair of CNNs, one to predict single attribute labels, and the second one to learn label co-dependencies conditioned on the image. For faces, Liu *et al*. [8] create the CelebA and the LFWA datasets, attribute labeled versions of the CelebFaces dataset [9] and the LFW dataset [10], and use cascaded CNNs to localize faces and predict their attributes in a weakly supervised manner.

Beyond simple attribute recognition, there has also been significant work in using attributes for auxiliary tasks. Vedaldi *et al*. [11] use attributes as a means of recognizing object parts and analyzing objects in detail. Both Duan *et al*. [12] and Lampert *et al*. [13] leverage attribute classification to perform fine-grained classification, with the work focusing on bird and animal classes respectively. Farhadi *et al*. [14] and Kulkarni *et al*. [15] use attributes as a means of generating more detailed captions of images. More recently, there has been work on using attributes to guide generative modeling of images with Yan *et al*. [16] training a conditional Variational Auto Encoder to generate faces possessing the attribute specified, such as smiling.

While our work focuses on a framework for visual attribute recognition, we avoid making any assumptions about the underlying problem domain and demonstrate strong performance on a general, large-scale object attribute dataset, with the intent that our framework can be readily adapted to any of the above problem domains without significant overhead.

1.2.2 Multi-Task Models

A recent paradigm for attribute recognition models has been the use of CNN based Multi-Task models. Multi-Task models are models that learn different tasks simultaneously in order to learn better representations that generalize across all the tasks [17][18][19], and as

Figure 1.1: *Example images from COCO Attributes*. These are sample images from the dataset with the labels underneath listing the objects as identified in the dataset. Each object is labeled with a set of attributes *e.g.*the dog has the attributes “standing”, “hairy”, “fluffy”, “soft” and “tame” for (a). Similarly for (b), the person is labeled “adult” and a car in the background is labeled with “metal / metallic” and “parked”.

such, along with transfer learning, are the dominant paradigm for training deep networks. In this vein, Abdulnabi *et al*. [20] train multiple CNNs, one for each attribute, whose output is fed into a shared linear layer that learns to aggregate the representations in order to provide accurate attribute recognition for clothes. However, this formulation is very resource heavy, especially for large scale attribute recognition where the number of attributes can easily be above 100. As an improvement, Hand and Chellappa [21] proposed using a single CNN backbone network to extract visual features and then split the network into multiple heads, one for each attribute group such that each head can predict an attribute, with an extra linear layer at the end to learn attribute dependencies. While this method focuses only on faces and does not leverage any type of transfer learning, this formulation has the adverse effect of requiring the model be retrained on the entire dataset each time a new attribute is added. In contrast, while our model is partly inspired by this model, our model has the advantage of being much more efficient and lightweight, and easily extensible to any number of attributes, while only requiring the new modules be trained on the relevant subset of data.

1.2.3 Training Paradigms

While most recent work has focused on developing powerful models to learn attribute prediction and label co-dependency, there have been a few efforts in proposing improved training paradigms to allow the model to learn better representations for the task of multi-label classification. Multi-label classification is the task of tagging the entire image with multiple relevant labels and is thus different from visual attribute recognition. For example, as shown in 1.1, an image of a park containing humans, dogs and trees would have the labels "Person", "Dog", "Tree", etc. These are valid labels for the image but do not constitute visual attributes of the individual objects, such as "Running", "Standing", etc. However, it is easy to see how models designed for multi-label classification could be adapted and leveraged for visual attribute recognition, and we consider these methods accordingly. Recent work by Wang *et al*. [22] illustrates the use of deep learning models for learning multi-label classifiers. In their formulation, they pose the problem of multi-label classification as a single-label path prediction problem where an LSTM is trained to embed a sequence of applicable labels each dependent on the image features from a CNN and the previously output label. During inference, they use beam-search on the LSTM to find the most likely sequence of labels given the image. This model assumes an ordering to the labels in that the sequence of labels for each training sample is ordered by popularity in the dataset. While we make use of an LSTM to learn label dependencies from each neural module by embedding all the module outputs into a single embedding, we do not make any such assumptions in our formulation, thus leading to easier extensibility.

Martins & Astudillo [23] defined a new activation function targeted towards multi-label classification which they term as SparseMax. This activation learns to output sparse probabilities rather than a distribution over various classes, with non-zero probability indicating the presence of a label. They further formulate this activation function as a loss function, termed as SparseMax loss, to be used to train networks end-to-end with backpropagation. They illustrate the performance of this loss in a multi-label problem setting for text data,

showing encouraging performance on the tasks of text label recognition and attention. Encouraged by this work, we propose two new loss functions which directly optimize for the nDCG and mAP metrics and report the results of our analysis in section 2.1.

Another approach to learning visual attributes effectively is to use Curriculum Learning [24]. Curriculum learning is a training paradigm of deep networks in which training samples are provided in a meaningful way or in some pre-defined order to help the network learn better. As discussed in Sarafianos *et al*. [25], curriculum learning is leveraged by splitting the training set based on visual attribute clusters where the similarity is measured using the Pearson Correlation Coefficient, then sequentially training on each cluster starting with clusters having the strongest intra correlation. This approach is applied to attribute recognition of various datasets of standing humans with impressive performance.

1.2.4 Extreme Multi-Label Learning

There exists a sub-category of work on attribute labeling where the attribute space if of the order of 1 million labels [26] [27]. This is important for problems such as ranking and recommendation systems, where the number of results is to the order of a few millions.

Liu *et al*. [28] were the first to apply deep learning based models to the task of attribute labeling of documents. The key ideas in their work is compressing the number of feature maps to a very small size in order to allow the model to be computationally feasible, and using a binary cross-entropy loss for per-attribute classification.

To the best of our knowledge, ours is the first work to leverage ideas from the area of extreme classification for visual attribute labeling.

CHAPTER 2

METHODOLOGY APPROACH

2.1 Analysis On COCO Attributes

In this section, we systematically analyze various recent developments in order to establish common baselines across them and compare them in a fair manner. For our common benchmark, we use the train and val split of the large scale COCO Attributes [1] as our training and test sets. COCO Attributes labels each object in the dataset with attributes. Each object in the dataset has a corresponding bounding box, originally curated for the task of object detection, thus we crop out the objects and feed it as input to our network to perform object level attribute recognition, similar to [1]. This gives us a training set consisting of 188,426 object instances each labeled with 204 object attributes. Similarly, the test set consists of 62,271 object instances labeled with 204 object attributes. We add context padding to each cropped object [29] to better model the surrounding context. We use the commonly used Mean Average Precision (mAP) as our metric of evaluation, by computing average precision over our test set for each attribute individually and then computing the mean over the precision scores.

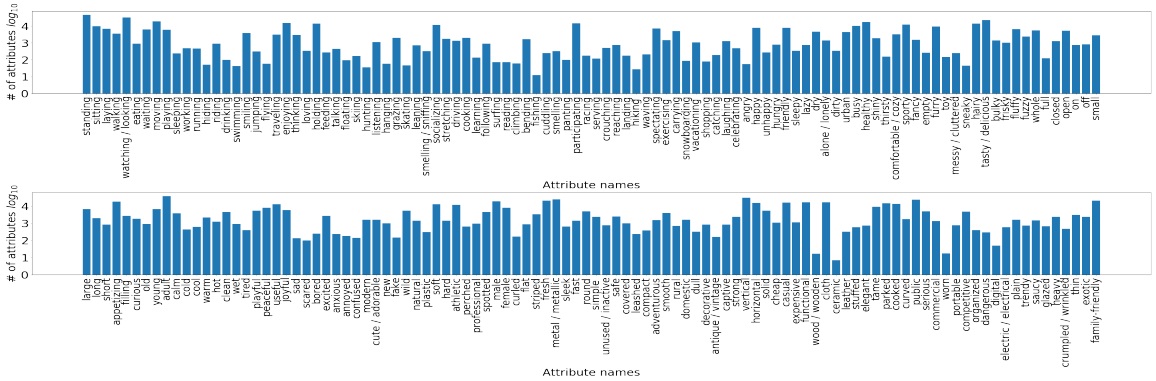

Figure 2.1: *COCO Attribute Frequences*. We visualize the per attribute frequencies on a log scale to illustrate which attributes are common and which are sparse.

2.2 COCO Attributes Baselines

Patterson & Hays [1] report three sets of baselines which we use as starting points for our analyses. The first baseline is random chance which is cited as 0.0514 mAP. The other two baselines use an AlexNet-based CNN [30] pretrained on both the ImageNet dataset and the Places dataset [31]. The first of these baselines trains individual SVM classifiers on the image features, with each classifier responsible for a single attribute, and reports a performance of 0.1395 mAP. The other baseline fine-tunes the CNN end-to-end using the Multi-Label Soft Margin Loss, and using the correlations learned from this loss, reports performance of 0.1574 mAP. Since we are primarily dealing with object images, we reproduce the second baseline using an AlexNet-based architecture pre-trained only on ImageNet.

2.3 Improved Architectures

Given the rapid progress of CNN architectures for ImageNet object classification in the past few years, the first question we can ask is whether using a more powerful CNN model can provide us with some initial performance gains. To this end, we experimented with various contemporary CNN models, all fine-tuned from ImageNet pre-training. For each model, we replaced the final fully connected layer with a 204-dimensional fully connected layer, initialized randomly. Each model was fine-tuned using the Multi-Label Soft Margin Loss on the entire training set, with the last fully connected layer trained with a learning rate an order of magnitude higher than the backbone model. The results of this experiment can be seen in table 2.1. All models were trained using the Adam optimizer [32] with an initial learning rate of 10^{-4} for 5 epochs. We use these hyperparameters for all subsequent experiments as well, noting exceptions where necessary.

As can be seen from the table, using deeper networks with more representational power provides us with higher performance. Interestingly, increasing the depth of the ResNet model does not give us a significant performance gain, thus we decide to limit our model

Table 2.1: Performance of different fine-tuned CNN architectures.

Architecture	mAP
Random	0.05137
AlexNet-SVM	0.13950
AlexNet	0.15444
VGG-19-BN	0.15467
ResNet50	0.1496610988
ResNet-101	0.16541
ResNet-152	0.16587

depth to ResNet-101. This also serves a practical purpose since using lightweight networks makes it easier to embed them in downstream applications.

2.4 Multi-Task Learning

From the previous subsection, we see how a simple architectural change can give us a significant boost in performance. However, these changes have been motivated from the task of object recognition and not attribute recognition. The next natural question to ask is, what further task-specific modifications can we make to the architecture to further boost performance? Abdulnabi *et al*. [20] first examined this problem by proposing a multi-task CNN which treats recognizing each attribute as an individual task. This models uses a different CNN for each attribute, with a common linear layer at the end that would take the feature vector of each CNN and learn the best output labels for the image. This scheme, however, suffers from being computationally very heavy, especially in cases where the number of attributes is much greater than the 23 present in the dataset used by them. Since COCO Attributes has 204 attributes, it is computationally infeasible to reproduce this model.

The use of Neural Modular Networks also constitutes a form of multi-task learning, since each module is learning a specific behavior. This is further examined in section 2.7.

2.5 Label Co-dependencies

One of the key insights from the last section illustrated that learning label co-dependencies for a set of binary classifiers is just as important as leveraging a common backbone network for the network to model attribute relationships well. This turns out to be an important component for performance if certain attributes are more sparse than others. The histogram in figure 2.1 illustrates the frequency of each attribute in the COCO Dataset, and it is evident that for data-driven models like the ones we have evaluated so far, modeling the under-represented attributes will be problematic due to the severe class imbalance. This motivates the need for us to further investigate techniques that employ learning not just prediction of the attributes but learning the relationships or co-dependencies between them as well.

One simple idea proposed by Kendall *et al*. [18] to tackle the class imbalance problem is to weigh the loss assigned to each label by a function of the number of samples belonging to that class/label. Abdulnabi *et al*. [20], and Hand & Chellappa [21] leverage the use of a single linear layer to learn the label relationships and boost performance. This idea has been approached in other works as well, especially in the area of multi-label classification. The work of Wang *et al*. [22] in particular explicitly tries to model this label co-dependency by tackling it as a sequential prediction problem with the use of an LSTM on the sequence of labels which is input to the model.

Our baselines also utilize a single linear layer in an effort to learn label co-dependencies, which is enforced via the use of the Multilabel Soft-margin loss. We further expand upon this idea in our proposed framework to explicitly model label co-dependencies to achieve substantial improvements in results.

2.6 Loss Formulations

Most loss functions in the literature focus on the task of single label classification (unimodal distributions) and given that attribute recognition is a task which requires multiple labels to be output (multimodal distribution), especially in a way that allows for the model to learn label co-dependencies, formulating a loss for attribute recognition is a challenging task. This is evident from the fact that most multi-label classification models and attribute recognition models either try to convert the problem at hand to a single label classification task [22] or use the standard multi-label soft margin loss as their model's loss function [1].

In an effort to provide a better loss function for multi-label classification, Martins & Astudillo [23] recently proposed the SparseMax activation as an activation layer that is capable of outputting sparse probabilities. This is in contrast to the Softmax activation or other activations where at no point in the output distribution is the value zero. Martins & Astudillo further derive a loss function based on the SparseMax activation, which they term as the SparseMax loss, to allow for training deep network end-to-end with backpropagation. We trained a baseline ResNet101 model with this loss only to discover that the model degenerates to outputting zeros everywhere. We hypothesize that this loss function is poorly suited to visual data and a high number of attribute labels.

Since our goal for training these models is to improve the mAP metric, it makes sense to utilize a loss function that optimizes for mAP. However, due to the sorting and ordering involved in these metrics, they possess discontinuities which make them unamenable to direct gradient based optimization techniques. Qin *et al*. [33] propose approximations for both the Normalized Discounted Cumulative Gain (**nDCG**) and Mean Average Precision (**mAP**) metrics, which allow for gradient-based methods to optimize over them. We implement these approximations in an efficient, batch-oriented manner and convert them to losses by subtracting the approximate values from 1, thus giving us two new losses, *nDCGLoss* and *MAPLoss*, which can be uses to train our deep visual attribute models in an

end-to-end fashion.

Both the nDCGLoss and the MAPLoss have a hyperparameter α which is used to approximate the ordering position. The MAPLoss has an additional hyperparameter β which is used to approximate the truncation function as described in [33]. We demonstrate results on these novel loss functions in section 4.2. To the best of our knowledge, our work is the first to optimize for nDCG and mAP directly in the domain of visual attribute labeling.

2.7 Attribute Module Network

Based on the above analyses, we now have a better understanding of how these various techniques operate on a common benchmark, with which we now present our novel framework for learning visual attribute recognition. We leverage an architecture similar to the MCNN-AUX architecture [21], in that our architecture splits the output of a base network into N heads, one for each attribute. However, instead of using a custom shallow network trained from scratch as the base network, we take advantage of the very deep, state-of-the-art ResNet101 model [34] pre-trained on ImageNet to extract image features from the image since this gives us better performance as per section 2.4. Furthermore, we propose using simple, uniform residual blocks as neural modules for each attribute and train each of these layers to output a binary value representing whether the attribute is present or not. This allows each module to learn an independent representation of the occurrence of an attribute using a binary cross-entropy loss.

Our design decision to use simple, uniform layers in contrast to complex custom modules for each attribute is two fold: (1) There are a fair number of attribute labels that are sparsely sampled in the dataset, thus having a high degree of parameterization could lead to overfitting and poor performance overall, and (2) as seen in [3], the use of simple and general modules allows the network to learn concepts quite well irrespective of the nature of the concept being learned.

With our network now outputting multiple attributes independently, we still need to

learn label co-dependencies. From section 2.5, we understand that learning attribute co-dependencies is essential for improved performance, especially to overcome the problem of sparse attribute labels. To enable learning of attribute co-dependencies, we use a simple linear layer over the sequence of outputs from each attribute, which learns to classify the sequence in order to perform joint recognition training on. We refer to the final classification layer together as the model head.

Due to the large number of attributes involved, computational memory limits need to be addressed. To this end, we leverage the ideas from Extreme Multi Labeling and down-project the feature channels before passing them onto the the attribute modules [28]. In our model, immediately after the backbone network gives us a high-level representation, we down-project the feature map (from the 2048 channels of a ResNet101 network) to a fixed size of 512 channels. For the attribute modules, we then down-project the feature maps to a fixed size of 128. We use 128 since it provides a good compromise between memory requirements and representational capacity.

To enforce each module to learn to predict an attribute label while also learning label co-dependencies, we formulate the training in a multi-task fashion. Each attribute module has a binary cross-entropy loss associated with it that encourages the module to do well on a single attribute. In addition to the gradients from the cross-entropy loss, the modules also receive gradients from the head which is trained jointly on all attribute labels using a suitable loss. This encourages the modules to learn to predict attributes based on the image features in a manner which is more consistent with other modules. We follow the same training scheme as in section 2.1.

Inference on our model follows the standard procedure of existing work: we pass the result of our model head through a sigmoid activation function to convert the logits values to probability values. We obtain a 204-dimensional vector as our final result with the index of the vector indicating which attribute is predicted.

CHAPTER 3

TECHNICAL DETAILS

In this section, we elaborate upon the approximations to the nDCG and mAP metrics as described in [33] that allow us to formulate them into differentiable loss functions.

3.1 Basic Definitions

The classical definitions of nDCG and mAP are given as

$$NDCG@k = N_k^{-1} \sum_{j=1}^{k} g(r_j) d(j) \tag{3.1}$$

where r_j denotes the relevance level (1 or 0 in our case), $g(r_j)$ denotes the gain function (*e.g.* $g(r_j) = 2^{r_j} - 1$), and $d(j) = 1/\log_2(1+j)$ denotes a discount function. N_k denotes the value of the ideal discounted cumulative gain *i.e.* when the ranking is perfect.

$$AP = \frac{1}{|D_+|} \sum_j r_j * Pre@j \tag{3.2}$$

where $Pre@j = \frac{1}{k} \sum_{j=1}^{k} r_j$ is a measure for evaluating the top k positions of a ranked list, $|D_+|$ denotes the number of relevant items (labels with 1) with respect to the query (the input image).

These measures can be reformulated with the use of a truncation function $\mathbf{1}(x)$ and a position function $\pi(x)$ instead giving us the following new formulations:

$$Pre@k = \frac{1}{k} \sum_{x \in L} r(x) \mathbf{1}\{\pi(x) \leq k\} \tag{3.3}$$

$$AP = \frac{1}{|D_+|} \sum_{x \in L} r_j * Pre@\pi(x) \tag{3.4}$$

$$NDCG@k = N_k^{-1} \sum_{x \in L} \frac{2^{r(x)} - 1}{\log_2(1 + \pi(x))} \mathbf{1}\{\pi(x) \leq k\} \tag{3.5}$$

The position function gives us a ranking of the elements in the list, the truncation function helps us evaluate a specific subset of the elements. L is the space of all labels.

The final definition we need is of a scoring function $s(x)$ which denotes the score a ranking function (*e.g.* a model) assigns to each label given the query/image.

3.2 Approximations

In our above reformulations, the position function π and the truncation function **1** are non-differentiable due to the existence of discontinuities caused by the sorting and ordering operations. In this section, we derive differentiable approximations of these functions.

3.2.1 Position Function Approximation

The position function can be redefined in terms of the scoring function:

$$\pi(x) = 1 + \sum_{y \in L, x \in L, y \neq x} \mathbf{1}\{s_{x,y} < 0\} \tag{3.6}$$

where $s_{x,y} = s_x - s_y$.

A natural way to approximate the position function is to approximate the indicator function **1** using a logistic function (similar to soft-argmax approximations):

$$\frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})} \tag{3.7}$$

where $\alpha > 0$ is a scaling constant. The higher the value of alpha, the more "exact" the approximation becomes.

Thus, we can approximate $\pi(x)$ as:

$$\hat{\pi}(x) = 1 + \sum_{y \in L, x \in L, y \neq x} \frac{\exp(-\alpha s_{x,y})}{1 + \exp(-\alpha s_{x,y})} \tag{3.8}$$

Note that nDCG does not require a truncation function since the set of labels is $\{0, 1\}$, thus we can replace π with $\hat{\pi}$ and our approximation is complete.

3.2.2 Truncation Function Approximation

Unlike nDCG, the AP (and consequently, mAP) measures requires the truncation function. In a similar vein to 3.2.1, we can approximate the truncation function using a slightly more involved logistic function:

$$\frac{\exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))}{1 + \exp(\hat{\pi}(y) - \hat{\pi}(x))} \tag{3.9}$$

where $\beta > 0$ is a scaling constant. The key idea here is that we can leverage the approximate position function to give us relative rankings which when scaled with the logistic function are either pushed to 0 or 1.

Thus the final approximation for AP becomes:

$$\hat{AP} = \frac{1}{|D_+|} \sum_{x \in L, y \in L} \left(\frac{r(y)}{\hat{\pi}(y)} + \sum_{x \neq y} \frac{r(y)r(x)}{\hat{\pi}(y)} \frac{\exp(\beta(\hat{\pi}(y) - \hat{\pi}(x)))}{1 + \exp(\hat{\pi}(y) - \hat{\pi}(x))} \right) \tag{3.10}$$

3.3 Loss Functions

Given our approximations, the loss functions are simply:

$$L(x, y) = 1 - \hat{NDCG}(x, y) \tag{3.11}$$

and

$$L(x, y) = 1 - \hat{AP}(x, y) \tag{3.12}$$

The implementation of these approximations with a matrix library, such as MATLAB, may seem non-trivial, but with the use of some key functions and tools provided with these libraries, the implementation becomes straightforward. We have released the accompanying source code for these loss functions to allow for easy reproducibility, possibly beyond the use case of attribute labeling.

CHAPTER 4

RESULTS

4.1 Empirical Results

4.2 Loss Results

In table 4.1 we show the results of training our basic models on the two losses in addition to the Multilabel soft-margin loss. We set the α and β values of the loss functions as 100 for all models, with the exception of Alexnet based models, where α and β are set to 1. This is done because due to the lack of Batch Normalization in Alexnet, the gradients explode to give NaNs, and thus smaller α, β values let ameliorate this issue at the cost of poorer approximations. This poorer approximation results in poorer mAP values for Alexnet based models compared to their multi-label soft margin loss variants, as seen in 4.1, highlighting the importance of good approximation of the positions of the rankings. For the ResNet based models, using these losses, in comparison to the multilabel soft-margin loss, improves the mAP significantly. These results validate the use of loss functions modeled after ranking metrics as better options for baseline models.

It is interesting to note that the results for deeper models are not necessarily better. We hypothesize that this is due to difficulty in gradient propagation for these loss functions since the output space upon which the losses operate is sparse. This would result in the gradient signal not being strong enough to sufficiently train larger networks. However, these results indicate the losses would be well suited to models which need to be light-weight and fit into a bigger system as a component.

Table 4.1: Summary of Results

Architecture	Loss	mAP
Random		0.05137
Base Alexnet + SVM		0.13950
Alexnet	Multilabel	0.15735
ResNet101	Multilabel	0.16541
ResNet152	Multilabel	0.16587
AlexNet	nDCG	0.13698
ResNet101	nDCG	0.18011
ResNet152	nDCG	0.17968
AlexNet	mAP	0.14278
ResNet101	mAP	0.17814
ResNet152	mAP	0.16921
Module + Linear	Multilabel	0.19600
Module + Linear	nDCG	0.17675
Module + Linear	mAP	0.18038
Module + Residual	Multilabel	0.19866
Module + Residual	nDCG	0.17915
Module + Residual	mAP	0.18008

4.3 Attribute Module Network Results

We train and evaluate our proposed Attribute Module Network in a similar fashion to the models in our analysis in section 2.1. We use two variants of the module network, one where each module is a stack of 3 linear layers with ReLU non-linearities, and another one which each module as a residual block [34]. Both variants are trained using the same hyperparameters to allow for a fair comparison.

On our test set, our model with the joint loss as the Multilabel soft-margin loss achieves state-of-the-art performance of 0.1960 mAP and 0.19866 mAP for the linear and residual variants respectively. However, the results using the nDCG and mAP losses, while improved over their corresponding non-module based networks, are not as impressive as using the Multilabel soft-margin loss. We hypothesize this is due to the complexity of the network which once again makes back-propagation of gradients harder, and the ranking-based losses have trouble with this. Investigating improvements to model architectures which can

better leverage these new losses would be an interesting direction for future work.

4.4 Precision-Recall Curves

We plot precision-recall curves of the predictions of our model and compare it to those of the baseline ResNet101 model (figure 4.1). For attributes with a few hundred to a few thousand samples, our model outperforms the baseline (attributes 57, 72 and 184 have 7253, 746, 1747 samples in the training set, respectively) indicating that our model is able to leverage the attribute label co-dependency to better predict sparse labels (attribute 0 has 45963 samples). However, as seen from the precision-recall curve of attribute 141, despite having 951 samples, there is still room for improvement.

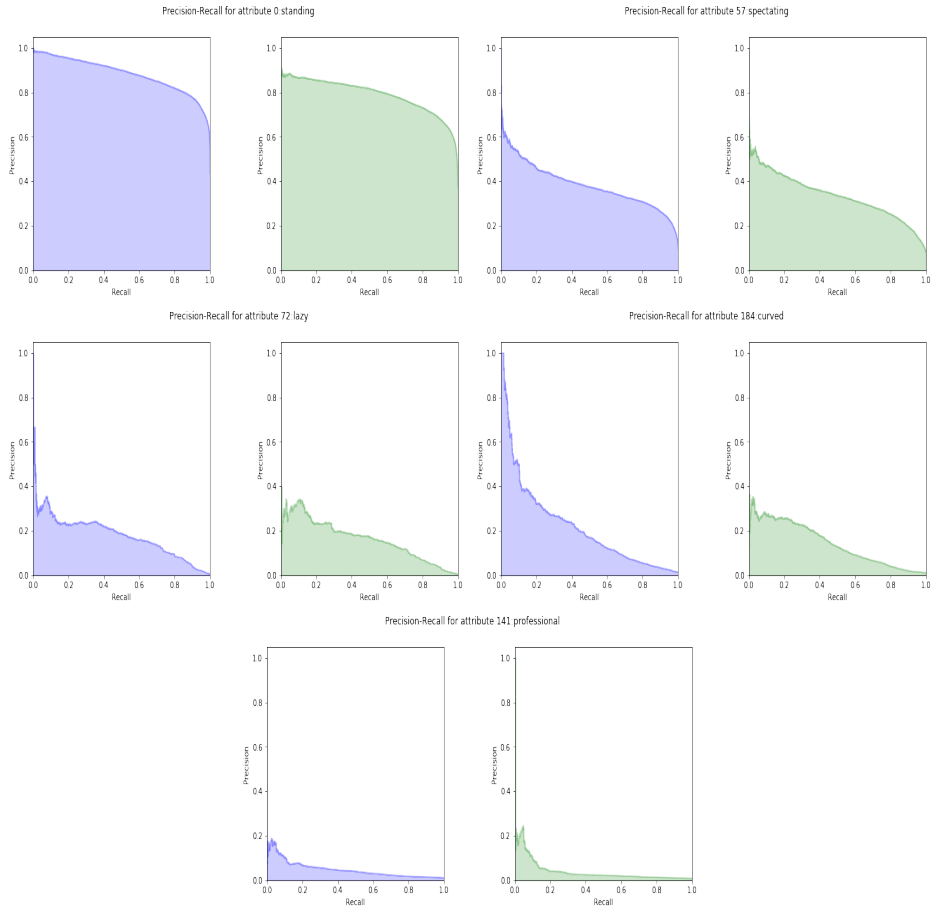

Figure 4.1: Precision Recall Curves of Attribute labels with different number of samples in the training set. Attribute Module Network predictions are represented as the purple curves, while the baseline ResNet101 model predictions are represented as green curves.

CHAPTER 5
CONCLUSION

We have presented a detailed analysis of various contemporary methods for attribute labeling on a challenging benchmark in order to fairly evaluate the strengths and weaknesses of each method. Furthermore, we have utilized the insights gleaned from this analysis to propose a novel and general framework for visual attribute labeling, including a new model architecture and new loss functions, which achieves state-of-the-art performance on the COCO Attributes dataset, is conceptually simple and easy to implement, and is easy to extend to more visual attribute categories.

To validate the efficacy of our proposed framework, we performed a series of empirical studies to investigate which parts of our framework are responsible for various performance boosts. While our proposed attribute module architecture gives us state-of-the-art results, it complexity may not be suitable to some tasks which require a lightweight attribute labeling mechanism. This is where the proposed loss functions shine, improving results of simpler, shallower models which can be easily and quickly trained and plugged into a bigger system.

Our hope is that the strong results demonstrated by our approach will not only help push forward the field of visual attribute labeling, but also encourage researchers to utilize our framework in unique and novel ways for problems where attribute labeling is required or may be helpful.

REFERENCES

[1] G. Patterson and J. Hays, "Coco attributes: Attributes for people, animals, and objects," *European Conference on Computer Vision*, 2016.

[2] J. Andreas, M. Rohrbach, T. Darrell, and D. Klein, "Neural module networks," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 39–48, 2016.

[3] J. Johnson, B. Hariharan, L. van der Maaten, J. Hoffman, L. Fei-Fei, C. L. Zitnick, and R. B. Girshick, "Inferring and executing programs for visual reasoning," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 3008–3017, 2017.

[4] O. Russakovsky, L. Fei-Fei, C. Li, Y. W. Li, and F. fei Li, "Attribute learning in large-scale datasets," in *ECCV Workshops*, 2010.

[5] H. Fukui, T. Yamashita, Y. Yamauchi, H. Fujiyoshi, and H. Murase, "Robust pedestrian attribute recognition for an unbalanced dataset using mini-batch training with rarity rate," *2016 IEEE Intelligent Vehicles Symposium (IV)*, pp. 322–327, 2016.

[6] J. Wang, X. Zhu, S. Gong, and W. Li, "Attribute recognition by joint recurrent learning of context and correlation," *2017 IEEE International Conference on Computer Vision (ICCV)*, pp. 531–540, 2017.

[7] D. Li, X. Chen, and K. Huang, "Multi-attribute learning for pedestrian attribute recognition in surveillance scenarios," *2015 3rd IAPR Asian Conference on Pattern Recognition (ACPR)*, pp. 111–115, 2015.

[8] Z. Liu, P. Luo, X. Wang, and X. Tang, "Deep learning face attributes in the wild," in *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[9] Y. Sun, Y. Chen, X. Wang, and X. Tang, "Deep learning face representation by joint identification-verification," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 1988–1996.

[10] G. B. Huang, M. A. Mattar, T. L. Berg, and E. G. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," 2007.

[11] A. Vedaldi, S. Mahendran, S. Tsogkas, S. Maji, R. B. Girshick, J. Kannala, E. Rahtu, I. Kokkinos, M. B. Blaschko, D. J. Weiss, B. Taskar, K. Simonyan, N. Saphra, and S.

Mohamed, "Understanding objects in detail with fine-grained attributes," *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3622–3629, 2014.

[12] K. Duan, D. Parikh, D. Crandall, and K. Grauman, "Discovering localized attributes for fine-grained recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[13] C. H. Lampert, H. Nickisch, and S. Harmeling, "Learning to detect unseen object classes by between-class attribute transfer," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 951–958, 2009.

[14] A. Farhadi, I. Endres, D. Hoiem, and D. A. Forsyth, "Describing objects by their attributes," *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1778–1785, 2009.

[15] G. Kulkarni, V. Premraj, S. Dhar, S. Li, Y. Choi, A. C. Berg, and T. L. Berg, "Baby talk: Understanding and generating image descriptions," 2011.

[16] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2image: Conditional image generation from visual attributes," in *ECCV*, 2016.

[17] R. B. Girshick, "Fast r-cnn," *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1440–1448, 2015.

[18] A. Kendall, Y. Gal, and R. Cipolla, "Multi-task learning using uncertainty to weigh losses for scene geometry and semantics," *CoRR*, vol. abs/1705.07115, 2017.

[19] I. Misra, A. Shrivastava, A. Gupta, and M. Hebert, "Cross-stitch networks for multi-task learning," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3994–4003.

[20] A. H. Abdulnabi, G. Wang, J. Lu, and K. Jia, "Multi-task cnn model for attribute prediction," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 1949–1959, 2015.

[21] E. Hand and R. Chellappa, *Attributes for improved attributes: A multi-task network utilizing implicit and explicit relationships for facial attribute classification*, 2017.

[22] J. Wang, Y. Yang, J. Mao, Z. Huang, C. Huang, and W. Xu, "Cnn-rnn: A unified framework for multi-label image classification," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2285–2294, 2016.

[23] A. Martins and R. Astudillo, "From softmax to sparsemax: A sparse model of attention and multi-label classification," in *Proceedings of The 33rd International Conference on Machine Learning*, M. F. Balcan and K. Q. Weinberger, Eds., ser. Proceed-

ings of Machine Learning Research, vol. 48, New York, New York, USA: PMLR, 2016, pp. 1614–1623.

[24] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*, ser. ICML '09, Montreal, Quebec, Canada: ACM, 2009, pp. 41–48, ISBN: 978-1-60558-516-1.

[25] N. Sarafianos, T. Giannakopoulos, C. Nikou, and I. A. Kakadiaris, "Curriculum learning of visual attribute clusters for multi-task classification," *CoRR*, vol. abs/1709.06664, 2017. arXiv: 1709.06664.

[26] Y. Prabhu, A. Kag, S. Gopinath, K. Dahiya, S. Harsola, R. Agrawal, and M. Varma, "Extreme multi-label learning with label features for warm-start tagging, ranking and recommendation," in *Proceedings of the ACM International Conference on Web Search and Data Mining*, 2018.

[27] A. Jalan and P. Kar, *Accelerating extreme classification via adaptive feature agglomeration*, 2019. arXiv: 1905.11769 [cs.LG].

[28] J. Liu, W.-C. Chang, Y. Wu, and Y. Yang, "Deep learning for extreme multi-label text classification," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '17, Shinjuku, Tokyo, Japan: ACM, 2017, pp. 115–124, ISBN: 978-1-4503-5022-8.

[29] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, ser. CVPR '14, Washington, DC, USA: IEEE Computer Society, 2014, pp. 580–587, ISBN: 978-1-4799-5118-5.

[30] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Commun. ACM*, vol. 60, no. 6, pp. 84–90, May 2017.

[31] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva, "Learning deep features for scene recognition using places database," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., Curran Associates, Inc., 2014, pp. 487–495.

[32] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. arXiv: 1412.6980.

[33] T. Qin, T.-Y. Liu, and H. Li, "A general approximation framework for direct optimization of information retrieval measures," *Inf. Retr.*, vol. 13, no. 4, pp. 375–397, Aug. 2010.

[34] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770–778, 2016.